

УДК 372.862

Е.С. ЛУКЬЯНОВА
(lukonya98@mail.ru)

Волгоградский государственный социально-педагогический университет

**ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ВИЗУАЛЬНОЙ СРЕДЫ “SCRATCH”
ПРИ ОБУЧЕНИИ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЮ
В ОСНОВНОЙ ШКОЛЕ***

Рассматриваются вопросы использования визуальных объектно ориентированных сред при обучении алгоритмизации и программированию учащихся основной школы. Основное внимание в работе уделяется использованию в образовательном процессе бесплатной визуальной среды “Scratch”. Представлен опыт использования данной среды в обучении программированию в 5–6 классах основной школы.

Ключевые слова: алгоритмизация, программирование, визуальная среда, Scratch, мотивация, обучение информатике, мультимедиа.

Сегодня в образовании прослеживается тенденция повышения значения обучению темам курса информатики, связанным с алгоритмизацией и программированием, что происходит из-за востребованности в современном информационном обществе специалистов в ИТ-индустрии и наукоемких предприятиях [3]. Раздел «Алгоритмизация и программирование» играет важнейшую роль в процессе развития интеллекта и других общих компетенций у школьников. К сожалению, настоящее положение этого раздела информатики незавидное: количество часов на изучение недостаточно, к тому же нельзя не отметить трудность восприятия учащимися сложных алгоритмических конструкций и основных понятий всего раздела. Невзирая на важность курса информатики и знаний, связанных с указанными темами, на сегодняшний день прослеживается снижение интереса у школьников к изучению программирования [3].

Анализ научно-педагогической литературы и педагогической практики показал, что большая часть учебно-методических пособий и разработок для обучения алгоритмизации и программированию предназначается в основном для 7–11 классов (базовый и профильный уровень) и для начальной школы (пропедевтический уровень). При обучении алгоритмизации и программированию в 5–6 классах перед учителями до сих пор существует проблема учебно-методической поддержки обучения по данным темам. Данную проблему обычно решают в двух аспектах: продолжить более углубленное изучение программных продуктов, рассматриваемых в начальной школе (например, «Алгоритмика», «Кумир», «Лого», «ПервоЛого» и др.) или начать изучать языки программирования, предназначенные для старших классов (например, “Pascal”, “VBA”, “Delphi” и др.) [8, 11]. Многие исследователи отмечают, что альтернативой являются визуальные среды программирования, которые не требуют высокопрофессионализма в области программирования, значительно сокращают время на разработку программы, позволяют учитывать возрастные особенности обучающихся [1, 3, 6, 7].

Под визуальными средами разработки компьютерных игр понимают игровые движки, которые являются центральными программными компонентами игр и обеспечивают основные технологии для запуска игр, взаимодействия с операционной системой и т. д., а также конструкторы игр, со встроенными игровыми движками и визуальным программным интерфейсом [7]. В данных средах, имеется возможность, например, с помощью визуального редактора создать игру без написания текста программы на языке программирования. Текст программы создает конструктор автоматически в визуальном редакторе, его можно при необходимости смотреть и править, что позволяет встроить такие визуальные среды программирования в учебный процесс [6, 8].

Разрабатывая самостоятельно компьютерные игры в визуальных средах, учащиеся приобретают и совершенствуют умения и навыки программирования в процессе игры, что значительно повышает

* Работа выполнена под руководством Куликовой Н.Ю., кандидата педагогических наук, доцента кафедры информатики и методики преподавания информатики ФГБОУ ВО «ВГСПУ».

мотивацию, когда они наделяют создаваемых персонажей игры определенными свойствами, программируют их действия, при этом используя основные алгоритмические конструкции, которые при традиционном обучении воспринимались бы очень сложно [2].

В качестве наиболее известных и популярных сред можно привести: Alice (www.okindiansbc.org); Kodu (<https://www.kodugamelab.com>), Scratch (<http://scratch.mit.edu>), LightBot (www.gameroo.nl/games/light-bot); StencylWorks (www.stencyl.com); Game Editor 1.40 (game-editor.com); Squeak (www.squeakland.org) и др. [2, 3, 8].

Особо отметим набирающую большую популярность в образовательном пространстве объектно ориентированную среду программирования “Scratch”, в которой программные блоки собираются из кирпичиков-команд, разделенных в группы по цвету и своему назначению соответственно.

Среда “Scratch” устроена следующим образом: в ней можно управлять разными объектами, легко их видоизменять, перемещать по экрану, задавать форму взаимодействия между ними и др. Главный принцип составления программ – собирание ее из графических блоков, находящихся в специальных разделах по их общему назначению [2, 8]. Необходимо отметить, что среда имеет все необходимые технические возможности: возможность объектноориентированного языка программирования высокого уровня, наличие графического редактора, встроенных инструментов для создания музыки, системы помощи пользователям, готовых проектов других пользователей и др.

Рассмотрим основные возможности “Scratch” и рекомендации учителю по созданию заданий учащимся. В данный момент среда не поддерживает процедуры, функции и рекурсию, поэтому обучение программированию в старшей школе было бы достаточно ограниченным. Однако технических характеристик среды достаточно для демонстрации теоретического материала, практических работ и других организационных форм урока по информатике для учащихся 5–6 классов.

Обучение программированию в школе важно проводить, используя примеры типовых задач, в которых постепенно усложняется структура алгоритмов. Это поможет систематизировать знания учащихся по классификации алгоритмов, необходимых в дальнейшем для программирования (линейные алгоритмы, ветвящиеся алгоритмы, циклические алгоритмы). Среда способствует связи всех теоретических знаний алгоритмов и программирования одной практической задачей – созданием компьютерной игры. Конечно же, в процессе создания игры учащимися легко осваиваются сложные для восприятия принципы объектно ориентированного программирования [4].

Необходимо обеспечить заранее ознакомление с целью учебной деятельности, создание необходимой познавательной мотивации у обучаемых. При решении проблемных задач учащимися применяется компьютерный эксперимент, вследствие которого формируются такие регулятивные универсальные учебные действия, как целеполагание, планирование, реализация плана, прогнозирование деятельности, контроль, коррекция [8, 9].

Среда позволяет обмениваться собственными разработками с участниками сообщества авторов Scratch-проектов во всем мире (scratch.mit.edu). Сообщество с открытым характером, т. е. в нем любой пользователь может не только посматривать все материалы, но и выкладывать на сервер сообщества собственный проект, зарегистрировавшись в нем [10].

На факультете математики, информатики и физики Волгоградского государственного социально-педагогического университета студентами проводится активная работа по исследованию образовательных возможностей компьютерных игр, визуальных объектно ориентированных сред программирования и технологий создания игр, а также рассмотрению вариантов использования визуальных сред программирования при смешанном обучении и др. [2, 3, 7]. В качестве примера подобной работы приведем лабораторно-практические задания для практической работы в 6 классе, разработанные автором в соответствии с методическими рекомендациями по изучению темы «Алгоритмика» с использованием среды “Scratch”.

В рабочей программе для 5–6 классов Л.Л. Босовой на изучение алгоритмов и начального программирования отводится 10 часов (3 часа теории и 7 часов практики) [1]. По окончании изучения теории раздела «Алгоритмика» и выполнения предложенных учителем заданий, учащиеся узнают,

что такое алгоритм и какая его роль в современных системах управления; познакомятся с основными свойствами алгоритмов, способами записи алгоритмов (блок-схемами, учебным алгоритмическим языком), основными алгоритмическими конструкциями (следование, ветвление, выбор (множественный), цикл, структуры алгоритмов) [1]. Разработанные задания можно использовать для формирования описанных выше результатов и при подготовке учащихся к программированию в более сложных и менее дружественных средах программирования в 8–9 классах.

Далее представлена одна из лабораторно-практических работ по данной теме.

Практическая работа «Создание скриптов (программ) в среде “Scratch”»

Цель: научиться создавать программы в “Scratch”, используя линейные алгоритмы, алгоритмы ветвления и циклы.

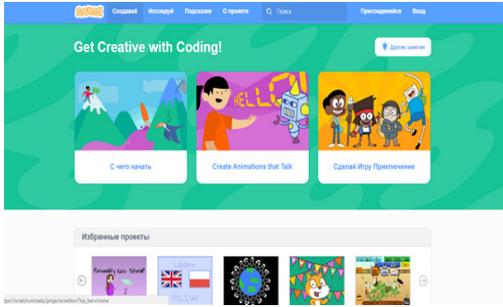
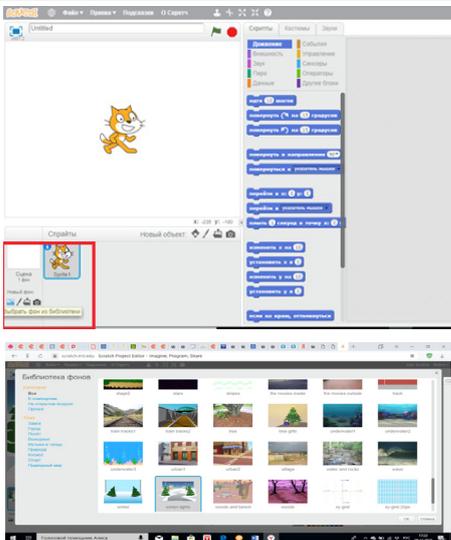
Задание 1: Создайте мини-мультфильм про дружелюбного пингвинёнка Рико, который добирается домой несколькими шагами по тропинке. Выполните следующий сценарий: при нажатии на флажок  Пингвин говорит «Привет!» и осуществляет поступательное движение по дороге (описано в технологии выполнения) несколькими шагами из координат (-70; -90) к координатам левого дома

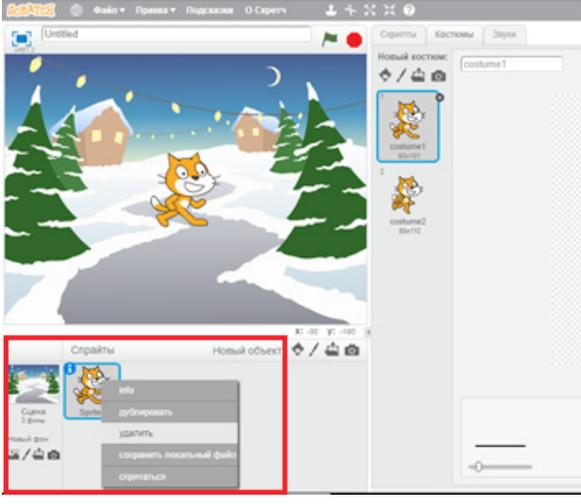
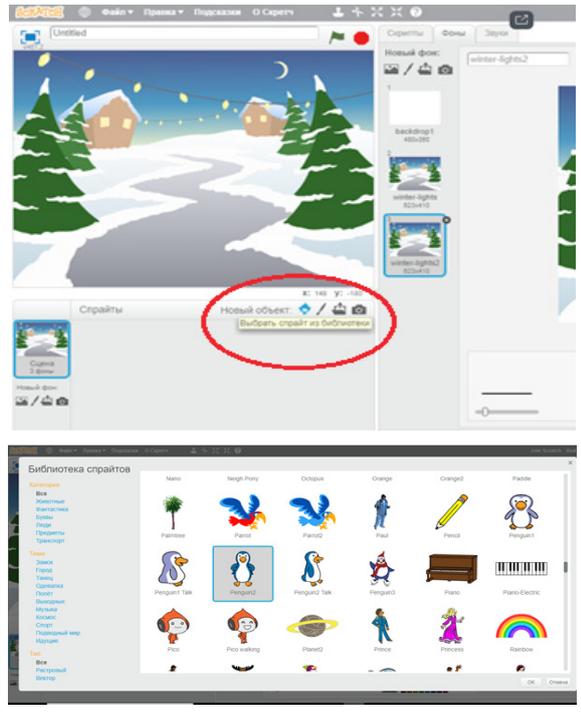


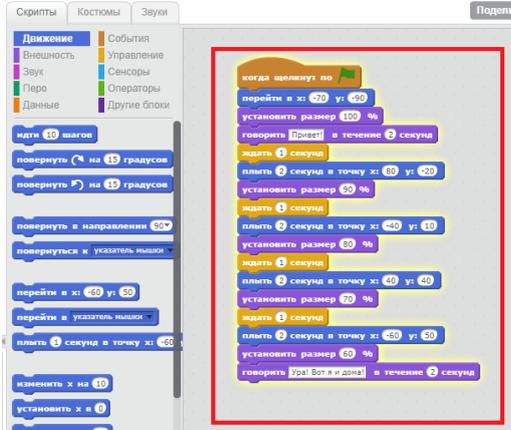
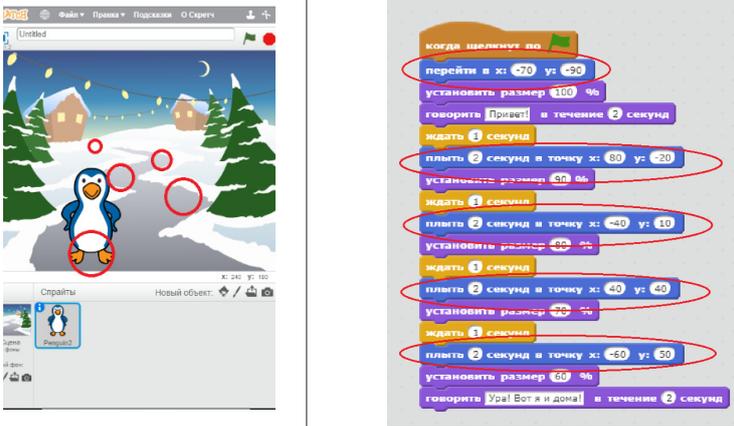
с помощью блока движения . С каждым перемещением размер спрайта должен уменьшаться на 10%, после чего он говорит «Вот я и дома! Ура!». Выполните задание в соответствии с указанной технологией его выполнения (табл. 1).

Таблица 1

Технология выполнения первого задания «Создание скриптов (программ) в среде “Scratch”»

Алгоритм выполнения задания	Примерные результаты в среде “Scratch”
1. Запустите среду “Scratch”, щелкнув по ярлычку на рабочем столе. Если вы выполняете задание не в классе то, открыв ссылку https://scratch.mit.edu/ (нажав CTRL и щелкнув по ссылке). 2. Нажмите на кнопку «Создавай» и перед нами откроется уже знакомое рабочее поле!	
3. Нажмите на кнопку «Выбрать фон из библиотеки», после чего выберите нужный нам фон из предложенных двойным щелчком мыши по нему.	

Алгоритм выполнения задания	Примерные результаты в среде “Scratch”
<p>4. Уберите Кота, выделив этот спрайт и нажав кнопку «Удалить» в контекстном меню.</p>	
<p>5. Добавьте нужный спрайт из предложенных (Пингвина), также щелкнув по нему двойным щелчком.</p>	
<p>6. Переместите Пингвина в начало дорожки Вашей сцены (фона). Для этого необходимо нажать на Пингвина левой кнопкой мыши и перенести его в нужное место.</p>	

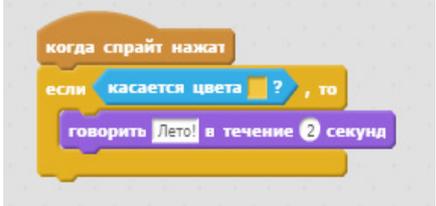
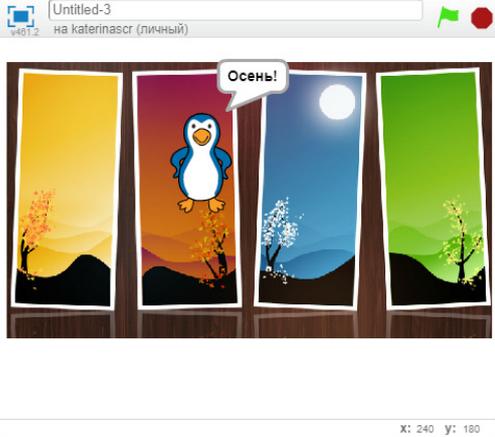
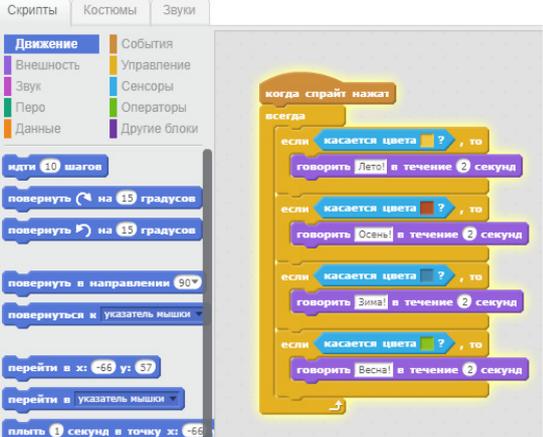
Алгоритм выполнения задания	Примерные результаты в среде “Scratch”
<p>7. В окне команд для спрайта Пингвин составьте следующую программу, используя область скриптов, которая находится в левой части окна.</p>	
<p>ПРИМЕЧАНИЕ! Синие блоки справа в сценарии обозначают координаты нового расположения Пингвина. Проверьте, все ли идет по сценарию и нажмите на зеленый флажок, чтобы посмотреть свой первый мини-мультик в “Scratch”!</p>	
<p>Дополнительное задание: измените готовый сценарий так, чтобы Пингвин говорил какую-нибудь фразу после каждого изменения положения.</p>	

Задание 2: Добавьте спрайт Пингвин, который при перемещении по сцене с изображением времен года, называет время года, на изображении которого он находится по щелчку на спрайте. Условие проверки времени года реализовать нужно с помощью алгоритмической конструкции «Если» и блоков «Сенсоры». Выполните задание в соответствии с указанной технологией его выполнения (табл. 2).

Таблица 2

Технология выполнения второго задания «Создание скриптов (программ) в среде “Scratch”»

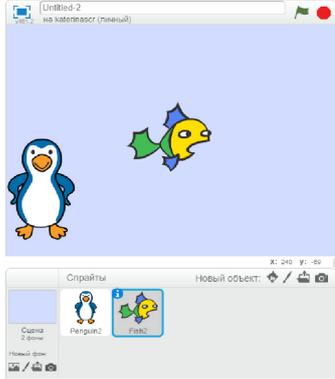
Алгоритм выполнения задания	Примерные результаты в среде “Scratch”
<p>1. Добавьте на место сцены изображение «временагода.jpg» из папки “Scratch” на Рабочем столе. Удалите спрайт Кот и добавьте спрайт Пингвин аналогично заданию № 1.</p>	

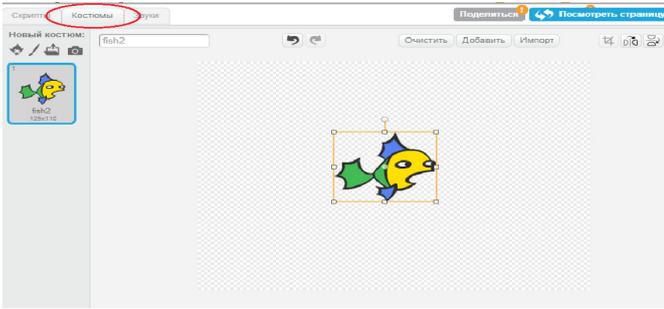
Алгоритм выполнения задания	Примерные результаты в среде “Scratch”
<p>2. Составьте из блоков конструкцию для проверки лета, чтобы при касании Пингвином жёлтого цвета возникала фраза «Лето!» в течении 2-х секунд.</p>	
<p>3. Составьте из блоков конструкцию для проверки остальных времен года, аналогично 2-му пункту для лета. Добавьте «всегда» для автоматической проверки условных циклов.</p>	<p>3. Составьте из блоков конструкцию для проверки остальных времен года, аналогично 2-му пункту для лета. Добавьте «всегда» для автоматической проверки условных циклов.</p>
	
<p>Дополнительное задание: создайте для каждого времени года свое «движение» Спрайта (Например, смена костюма или мини-анимация)</p>	

Задание 3: Добавьте спрайт Рыба в центре сцены и спрайт Пингвин, для которого создайте следующий сценарий: Пингвин в нижнем левом углу говорит «Привет! Где мой обед?», далее изменяет случайным образом своё расположение на сцене, пока не касается спрайта Рыба (касание обозначить с помощью блоков «Сенсоры»), после чего Пингвин говорит «Рыбка!». Выполните задание в соответствии с указанной технологией его выполнения (табл. 3).

Таблица 3

Технология выполнения третьего задания «Создание скриптов (программ) в среде “Scratch”»

Алгоритм выполнения задания	Примерные результаты в среде “Scratch”
<p>1. Измените сцену на любой понравившийся однотонный фон и удалите спрайт Кот (правила выполнения этих действий описаны в задании № 1). Добавьте спрайт Пингвин и Рыба (fish2).</p>	

Алгоритм выполнения задания	Примерные результаты в среде «Scratch»
<p>2. Нажав на вкладку «Костюмы» в правом окне, открой редактирование спрайта Рыба. Уменьши ее размер примерно в 2 раза регулируемыми размер стрелками.</p>	
<p>3. Переместите Пингвина в нижний левый угол сцены. Создайте из блоков следующий скрипт для этого спрайта</p>	
<p>4. Сделайте так, чтобы после фразы «Рыбка» прозвучал какой-либо звуковой сигнал на Ваше усмотрение.</p>	
<p>Дополнительное задание: с помощью блока «отправить сообщение 1» в скрипте спрайта Пингвин и события  в скрипте Рыбы сделайте так, чтобы Рыба говорила фразу «Не ешь меня!».</p>	

Разрабатывая собственные компьютерные игры в визуальной среде, учащиеся получают и совершенствуют умения и навыки программирования с высокой мотивацией в процессе игры, когда они наделяют создаваемых персонажей игры определенными свойствами, программируют для них действия, при этом используют базовые алгоритмические конструкции, которые при традиционном обучении они воспринимались ими очень сложно.

Отметим, что при составлении практических работ на «Scratch» учителю рекомендуется делать опору на творческий потенциал ученика, давать ученику определенную свободу в создании скрипта или сцены, чтобы обеспечить развитие его способностей в плане познания и творчества. Подведение учителем к проблеме – очень важный момент при изучении программирования и может проследиваться в логике лабораторных работ на «Scratch». Например, содержащие последовательные повторяющиеся действия предполагают подведение учащихся к использованию циклов. Необходимость исследовать явления, объекты, которые описываются с помощью таких процессов в проблемных задачах, требует освоения инструментария среды.

Подводя итоги, отметим, что «Scratch» является довольно новой, но уже одной из популярных в образовании сред визуального программирования. Ее визуальные возможности можно наилучшим образом использовать при обучении школьников темам «Алгоритмизация и программирование».

В настоящее время по всему миру наблюдается активное внедрение в образовательные программы общеобразовательных школ учебных предметов научного, технологического, инженерного и ма-

тематического направления. К слову, что будущее – за технологиями, сейчас обсуждаемой проблемой стало изучение школьного курса информатики, в частности вопросов обучения программированию в школе. Исследователи, рассматривающие программирование в качестве способа общения с компьютером на понятном ему языке, всегда поддерживали идею обучения учащихся программированию с раннего возраста [5]. С помощью такой среды, как “Scratch”, у учителей информатики появляется возможность организовать обучение алгоритмизации и программированию качественно, доступно и легко на практических примерах, повысить при этом уровень мотивации учащихся к обучению за счет использования различных мультимедийных возможностей среды и игровой компоненты.

Литература

1. Босова Л.Л., Босова А.Ю. Информатика. Программа для основной школы. 5–6 классы. 7–9 классы. М.: БИНОМ. Лаборатория знаний, 2015.
2. Быкова А.Р. Использование среды Scratch для обучения программированию учащихся основной школы // Наука молодых – будущее России: сб. науч. ст. II Междунар. науч. конф. перспектив. разработок молодых ученых: в 5 т. (г. Курск, 13–14 дек. 2017 г.). Курск: Университетская книга, 2017. С. 69–72.
3. Данильчук Е.В., Куликова Н.Ю., Гермашев И.В. Методические особенности формирования готовности будущего учителя информатики к разработке и использованию компьютерных игр в обучении алгоритмизации и программированию // Изв. Волгоград. гос. пед. ун-та. 2018. № 5(128). С. 42–49.
4. Жемчужников Д.Г. Создание компьютерных игр как средство обучения школьников программированию // Информатика и образование. 2012. № 8(237). С. 49–51.
5. Жилбаев Ж.О., Мукашева М.У., Шуиншина Ш.М. Программирование в школе: исследование отношения и потребностей субъектов образовательного процесса // Новые информационные технологии в образовании и науке: сб. тр. X Междунар. науч.-практ. конф. (г. Екатеринбург, 27 февр.–3 марта 2017 г.) Екатеринбург: Рос. гос. проф.-пед. ун-т, 2017. С. 27–32.
6. Карпенко О.М., Лукьянова А.В., Абрамова А.В. [и др.] Геймификация в электронном обучении // Дистанционное и виртуальное обучение. 2015. № 4(94). С. 28–43.
7. Куликова Н.Ю. Создание и использование интерактивных компьютерных игр как средство активизации познавательной деятельности обучающихся на уроках информатики // Современные информационные технологии в образовании: сб. материалов XXVIII Междунар. конф. (Троицк-Москва, 27 июня 2017 г.). М.: Москов. издат.-полиграф. колледж им. И. Федорова. 2017. С. 27–29.
8. Лукьянова Е.С. Использование визуальной среды Scratch в обучении алгоритмизации и программированию // Научный руководитель. 2018. № 6(30). С. 25–33.
9. Макарова Н.В., Нилова Ю.Н. Методика формирования навыков программирования и моделирования // Информатика и образование. 2014. № 2(251). С. 29–32.
10. Патаракин Е.Д. Школа Scratch // Школьные технологии. 2010. № 4. С. 132–135.
11. Храмова М.В., Феоктистова О.А. Использование языка Scratch в курсе теории и методики обучения информатики // Вестник Московск. гор. пед. ун-та. Сер.: Информатика и информатизация образования. 2008. № 16. С. 179–181.

EKATERINA LUKYANOVA

Volgograd State Socio-Pedagogical University

PECULIARITIES OF USE OF VISUAL ENVIRONMENT “SCRATCH” WHILE TEACHING OF ALGORITHMIZATION AND COMPUTER PROGRAMMING IN MIDDLE SCHOOL

The article deals with the issues of using visual object oriented environments while teaching of algorithmization and computer programming of students in middle school. There is paid special attention to the usage of free visual environment “Scratch” in educational process. There is demonstrated the experience of the environment while teaching computer programming in 5–6 grades of middle school.

Key words: algorithmization, computer programming, visual environment, Scratch, motivation, teaching of Computer Studies, multimedia.